Variant and Variability Management with

Dr. Danilo Beuche
danilo.beuche@pure-systems.com
Who Needs Product Lines?

SIEMENS VDO Automotive

Audi
Porsche
Citroën
BMW
Mercedes-Benz
About pure-systems

- **Focus:** Support for Development of Variant Rich Systems
- **Business Areas**
  - Development Tools
  - Software Development
  - Consulting & Professional Services
  - Training
- **Customer**
  - Mainly embedded systems manufacturers
- **Founded 2001, Location Magdeburg, Germany**
### The Version Hell

<table>
<thead>
<tr>
<th>Component</th>
<th>Product 1</th>
<th>Product 2</th>
<th>Product 3</th>
<th>Product 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component A</td>
<td>1.0</td>
<td>1.1</td>
<td>1.3</td>
<td>2.0</td>
</tr>
<tr>
<td>Component B</td>
<td>1.0</td>
<td>1.2</td>
<td>2.1</td>
<td>2.4</td>
</tr>
<tr>
<td>Component</td>
<td>1.0</td>
<td>1.0</td>
<td>2.3</td>
<td>4.0</td>
</tr>
</tbody>
</table>
Orthogonality of Variants and Versions

Variants

<table>
<thead>
<tr>
<th>Mercedes-Benz</th>
<th>BMW</th>
<th>Citroen</th>
<th>Porsche</th>
<th>Audi</th>
</tr>
</thead>
</table>

Version / Time

- v1.0
- v1.2
- v1.3
- v1.4
- v2.0
- v2.2

© pure-systems GmbH 2008
**ALM and Variant Management**

**Variant Management with pure::variants**

<table>
<thead>
<tr>
<th>pure-systems</th>
<th>Portfolio Management</th>
<th>Requirement Management</th>
<th>Design and Development</th>
<th>Configuration Management</th>
<th>Quality Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Borland</td>
<td>Tempo</td>
<td>Caliber RM</td>
<td>Together</td>
<td>StarTeam</td>
<td>Silk</td>
</tr>
<tr>
<td>IBM</td>
<td>Portf. Manager</td>
<td>RequisitePro</td>
<td>SW Architect</td>
<td>ClearCase</td>
<td>ClearQuest</td>
</tr>
<tr>
<td>Telelogic</td>
<td>Focal Point</td>
<td>DOORS</td>
<td>TAU/Rhapsody</td>
<td>Synergy</td>
<td>Tester/Change</td>
</tr>
<tr>
<td>MKS</td>
<td>Portfolios</td>
<td>Requirements</td>
<td></td>
<td>Source</td>
<td>Test Mgmt.</td>
</tr>
</tbody>
</table>

**Application Lifecycle Management (ALM)**
Integration into Development Processes

- keep and use existing code base and tool environment
- version management support
- independent from technology, applicable to HW and SW based systems

Efficient Variant Modelling

- accumulation of configuration knowledge
- validation of variant configurations

Automated and Resource-Efficient Variant Generation

- source code packaging (e.g. from version management repositories)
- generation of code, documents, bills of material
Development with Variant Management

Problem Space
Collection of Features and Relations
Feature Model

Solution Space
Collection of Family Elements
Family Model

Single Problem
Desired/Required Features
Variant Model

Single Solution
Variant
Variant Realization

© pure-systems GmbH 2008
## Example Weather Station: Product Variants

<table>
<thead>
<tr>
<th>Variant</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermometer:</td>
<td>Display, Temperature</td>
</tr>
<tr>
<td>Indoor:</td>
<td>Display, Temperature, Pressure</td>
</tr>
<tr>
<td>Outdoor:</td>
<td>Display, Temperature, Pressure, Wind</td>
</tr>
<tr>
<td>Deluxe:</td>
<td>+ PC Data Recording</td>
</tr>
<tr>
<td>Internet Edition:</td>
<td>+ TCP/IP</td>
</tr>
<tr>
<td>PC Interfaces:</td>
<td>+ Serial Interface RS232</td>
</tr>
<tr>
<td></td>
<td>+ USB Interface</td>
</tr>
</tbody>
</table>
Problem Space – Feature Model

Output

WeatherMon

PCConnection
Conflicts: 'Trace'

Display

RS232Line

USBLine

DebuggingSupport

Trace

partial view
Example Weather Station: Hardware

- **µController (AVR)**
  - 4kB RAM, 8kB Flash

- **Sensors**
  - Wind
  - Pressure
  - Temp

- **Communication**
  - I²C
  - USB
  - RS232

- **Output**
  - Display

- **Data Transfer**
  - UDP/IP or Plain Text over USB/RS232
Solution Space – Family Model

WeatherStationSolutions

- Trace
  - hasFeature('Trace') and conflictsFeature('PCConnection')

- Display
  - hasFeature('Display')

- PCLine
  - hasFeature('Trace')
  - hasFeature('PCConnection')

- ps:class: RS232Line
  - hasFeature('RS232') or hasFeature('Trace')

- ps:class: USBLine
  - hasFeature('USBLine')

partial view
The Matrix view enables quick overview for multiple variants. Here the selection is shown (default visualization).

Each column represents a variant, rows are selectable elements like features.

Other visualizations can show variant specific defect and test states.
Product Integration

Development Tools

- DOORS
- Simulink
- ClearCase
- Codegenerators

Generic Data → Variability Data → Variant Data

© pure-systems GmbH 2008
Extensions integrate `pure::variants` optimally into existing tool chains.

Synchronizer for DOORS
Synchronizer for CaliberRM
Connector for MATLAB / Simulink
Connector for Source Code Management
Connector for Version Control Systems
Connector for SAP
Connector for ClearQuest
Connector for ClearCase
Connector for Bugzilla
Connector for Reporting with BIRT
Simulink Model with Variation Points
pure::variants based Simulink Configurator
The test state visualization shows for all selected features the success of the related tests (indicated by different icons).

It is visible here that not all variants have the same set of tests.
The defect state visualization shows for all selected features the existence of the related open defects (indicated by different icons).

It is visible here that not all variants have the same set of defects.
Product Variants

pure::variants Professional

- models are stored locally in file system as XML
- collaboration using standard software configuration management tools such as CVS, Subversion, Perforce, ...
- versioning and branching handled by software configuration management tool

pure::variants Enterprise

- models are stored in a centralized database
- collaboration in real-time, changes will be automatically visible to any connected user
- integrated reporting and history
- versioning and branching handled by pure::variants server
- all functionalities of p::v Professional
pure::variants - Integration as Key Concept

**p::v Eclipse Client**
- p::v Eclipse Plugins
- p::v Java Core

**p::v Server**
- p::v Runtime System

**Eclipse API**
- Eclipse Extension Points

**Java API**
- Java Interfaces

**SOAP Interface**

**C++ API**
- DLL/COM/OLE
**Problem**

- For 4 market segments control software is provided by different, worldwide distributed teams (about 70 persons). Software is based on very similar hardware.
- Reuse mostly „ad-hoc“.
- Cost and development effort were high.

**Task**

- Migration to controlled and systematic reuse based on common software plattform.
Our Role

- Participation during process definition
- Coaching and evaluation of newly setup platform team
- Supporting the migration to pure::variants

Results

- 1\textsuperscript{st} stage of migration finished successfully after 6 month: all projects develop based on common platform
- 2\textsuperscript{nd} stage of migration (introduction of pure::variants) almost completed
- Double number of products with about the same development capacity
Example Automotive Supplier

Problem

− Configuration of control software with about 2000 features could not be handled efficient manually.

Task

− Modelling of complete configuration knowledge in pure::variants and generation of configuration files.

− Distributed real-time remote access to models and configurations.
Example Automotive Supplier

Our Role

- Product supplier
- Adaptation of pure::variants to customer demands: integration of customer specific rule language.

Results

- Configurations are created in minutes instead of days.